

시본 (Seaborn) 플롯리 (Plotly)



소프트웨어융합대학원
진혜진

목차

👉 시본

👉 플롯리



시본



02 시본

1. 시본의 기본

- 시본(seaborn) : 맷플롯립을 바탕으로 다양한 함수 사용을 돕는 일종의 래퍼(wrapper) 모듈

```
import seaborn as sns
```

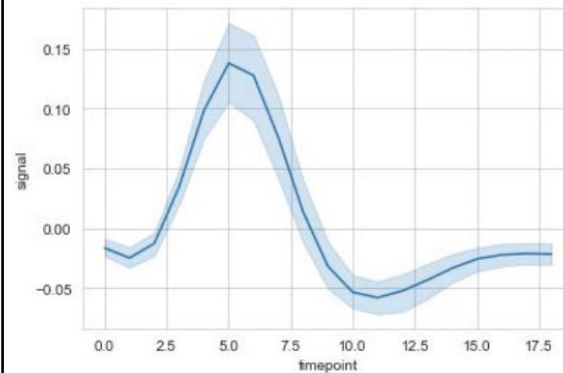
- 맷플롯립과 동일한 결과물이 나오며, 작성 과정이 간단
 - 그림 객체나 축 객체 같은 복잡한 개념이 없음
 - xticks 설정하지 않아도 각 축에 라벨 자동으로 생성
 - 데이터프레임과 x, y에 해당하는 열 이름만 지정하면 됨

02 시본

```
In [1]: import numpy as np          # numpy 모듈 호출
import pandas as pd              # pandas 모듈 호출
import matplotlib.pyplot as plt  # matplotlib 모듈 호출
import seaborn as sns            # (1)seaborn 모듈 호출

fmri = sns.load_dataset("fmri")  # (2)fmri 데이터셋 사용
sns.set_style("whitegrid")       # (3) 기본 스타일 적용
sns.lineplot(x="timepoint", y="signal", data=fmri)
                                   # (4) 선그래프 작성
```

Out [1]:



02 시본

- fmri 데이터는 연속형 값 외에도 다양한 범주형 값 가짐
 - 이럴 때 맷플롯으로 표현하기는 상당히 복잡하고,
시본은 hue 매개변수만 추가하면 그래프 그릴 수 있음

In [2]:

fmri.sample(n=10, random_state=1)

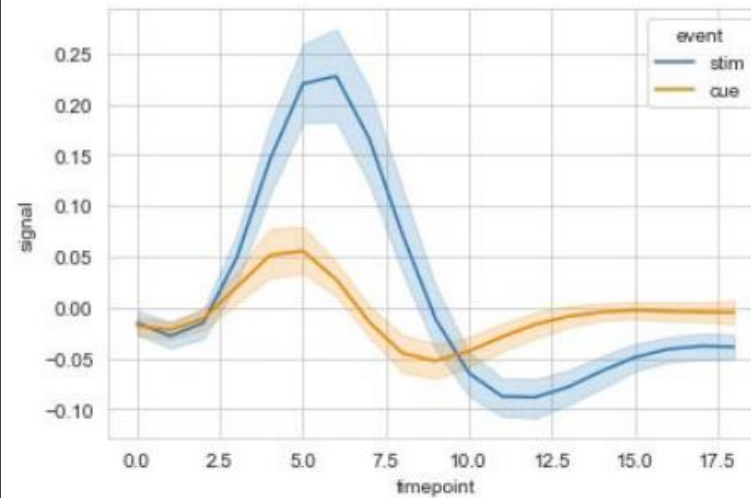
Out [2]:

	subject	timepoint	event	region	signal
806	s6	18	cue	parietal	0.019532
691	s5	15	cue	frontal	-0.019507
148	s5	8	stim	parietal	0.006805
676	s13	0	cue	parietal	-0.018394
156	s11	7	stim	parietal	0.254042
27	s1	17	stim	parietal	-0.038021
200	s11	4	stim	parietal	0.087175
262	s3	0	stim	parietal	-0.008576
94	s4	12	stim	parietal	-0.090036
339	s4	5	stim	frontal	0.455575

02 시본

In [3]: `sns.lineplot(x="timepoint", y="signal", hue="event", data=fmri)`

Out [3]:



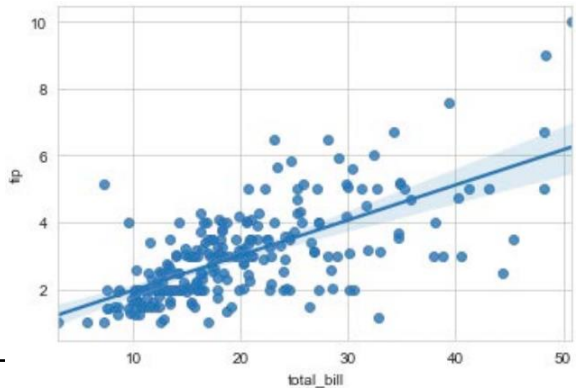
02 시본

2. 시본에서 사용하는 그래프

2.1 회귀 그래프

- 회귀 그래프(regression plot) : 회귀식을 적용하여 선형회귀 추세선을 그래프에 함께 작성
 - 선형회귀 추세선 : 데이터를 기반으로 데이터의 x값 대비 y값 변화를 예측하는 직선
- 함수 regplot 사용

02 시본

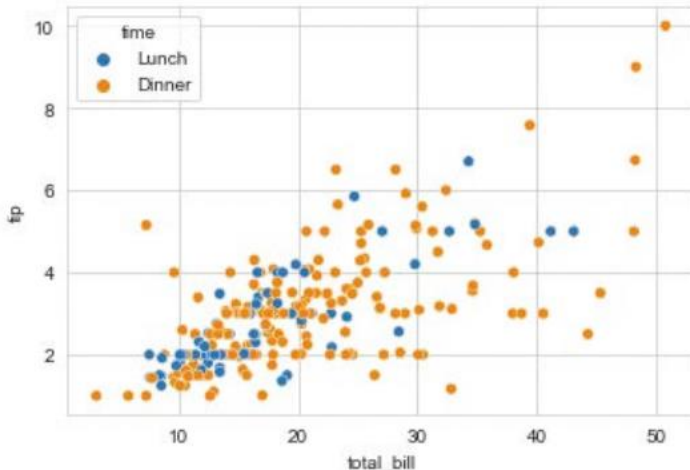
In [4]:	<pre>tips = sns.load_dataset("tips") sns.regplot(x="total_bill", y="tip", data=tips, x_ci=95)</pre>
Out [4]:	

- 매개변수 x_ci 는 신뢰구간의 비율을 나타냄

02 시본

2.2 산점도

- 산점도(scatter plot) : x, y를 기준으로 데이터의 분포 표현
- 함수 scatterplot 사용

In [5]:	<pre>tips = sns.load_dataset("tips") sns.scatterplot(x="total_bill", y="tip", hue="time", data=tips)</pre>
Out [5]:	

02 시본

2.3 비교 그래프

- 비교 그래프(counter plot) : 범주형 데이터의 항목별 개수

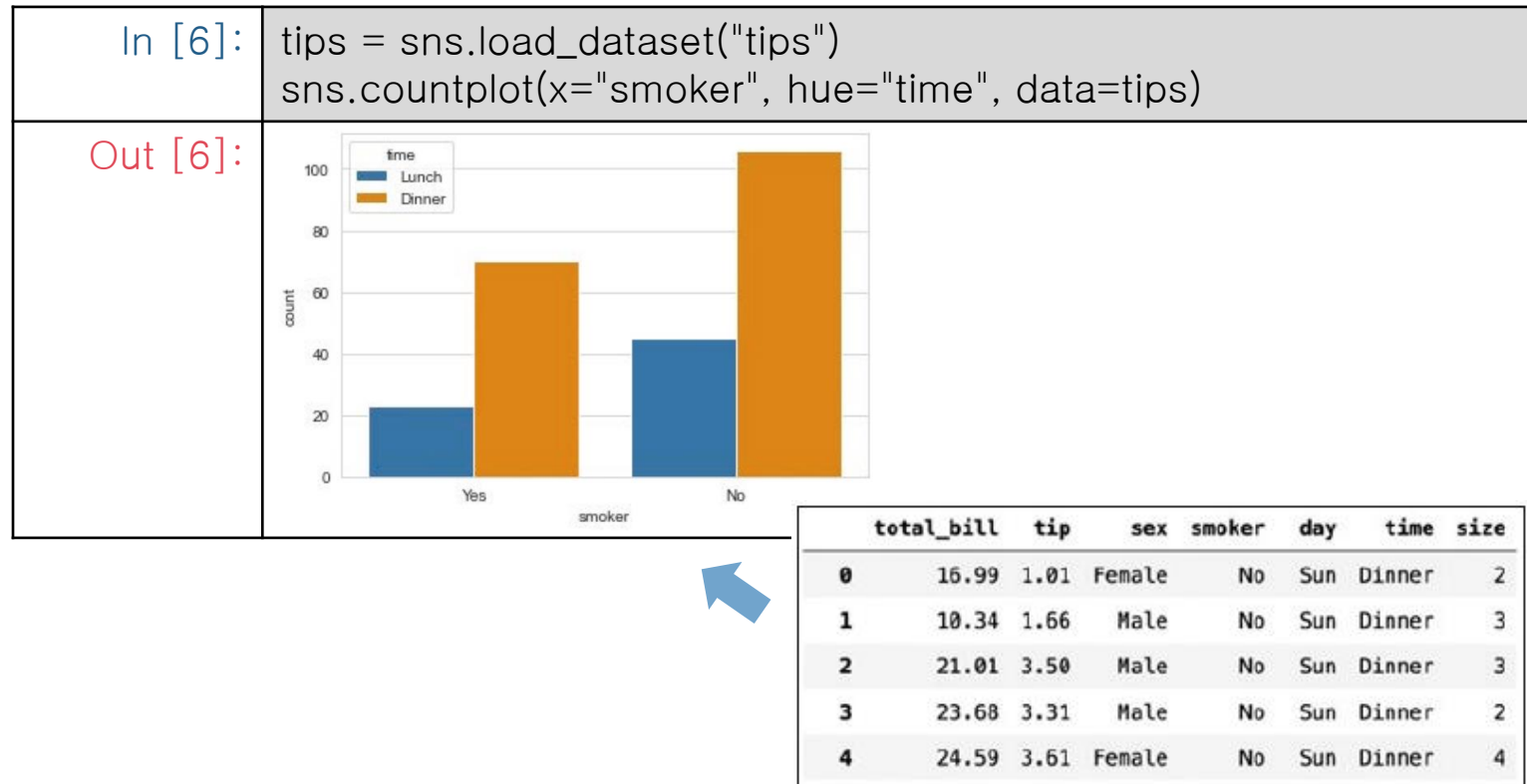
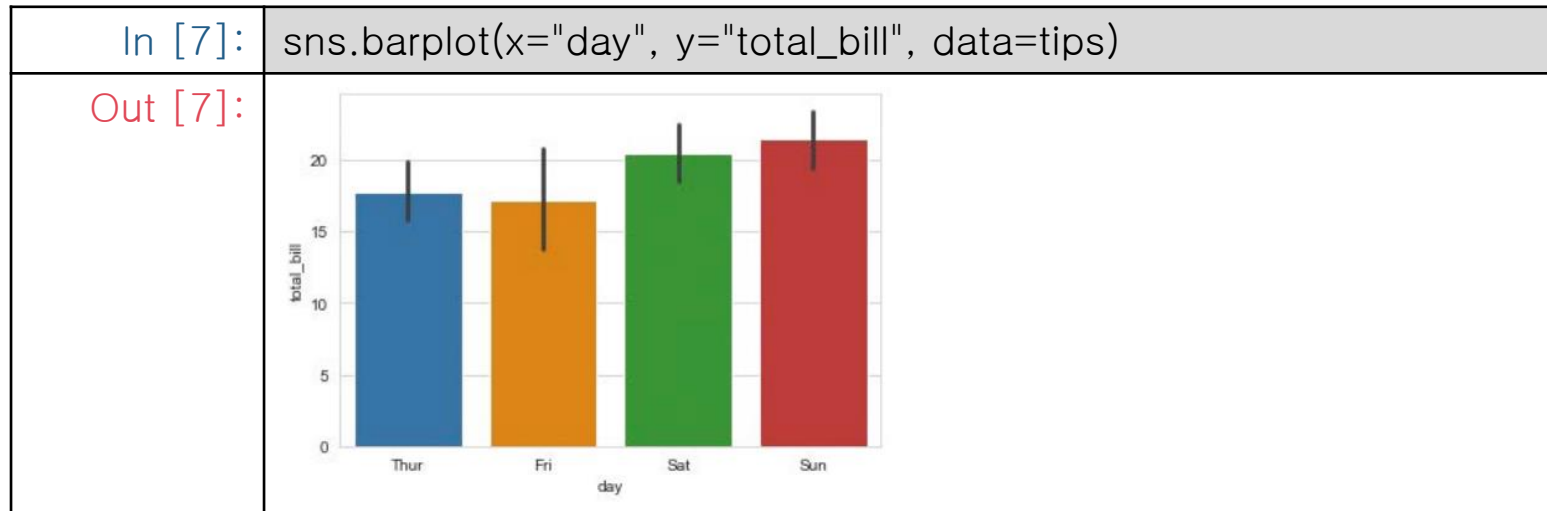


그림 5-4 비교 그래프 작성을 위한 데이터

02 시본

2.4 막대그래프

- y 값이 연속형 값일 경우 해당 값들의 평균을 나타냄
- 데이터의 신뢰구간을 검은색 막대로 표현
- 함수 barplot 사용



02 시본

3. 사전 정의된 그래프

- 맷플롯립 관점에서 여러 그래프들을 합쳐 정보를 추출
- 특히 범주형 데이터에 유용

3.1 분포를 나타내는 그래프 : 바이올린 플롯과 스웜 플롯

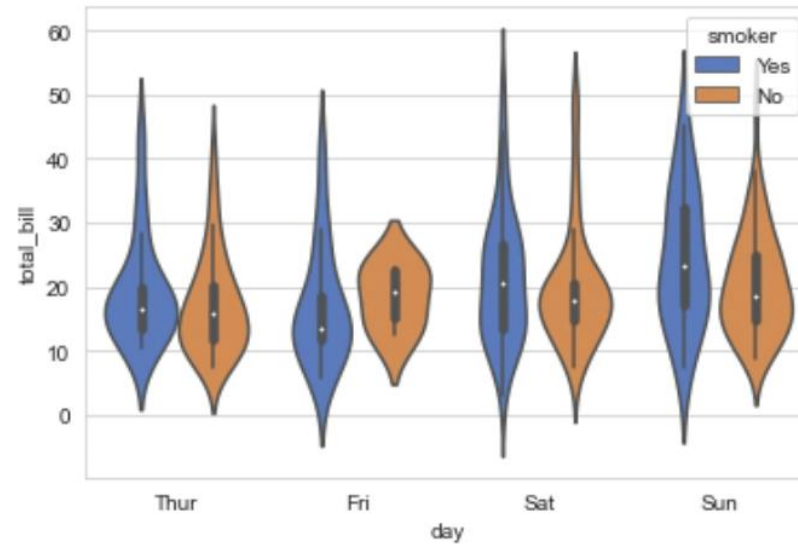
- 바이올린 플롯(violin plot) : 상자그림과 분포도를 한 번에 나타낼 수 있음
 - x축에는 범주형 데이터, y축에는 연속형 데이터

02 시본

In [8]:

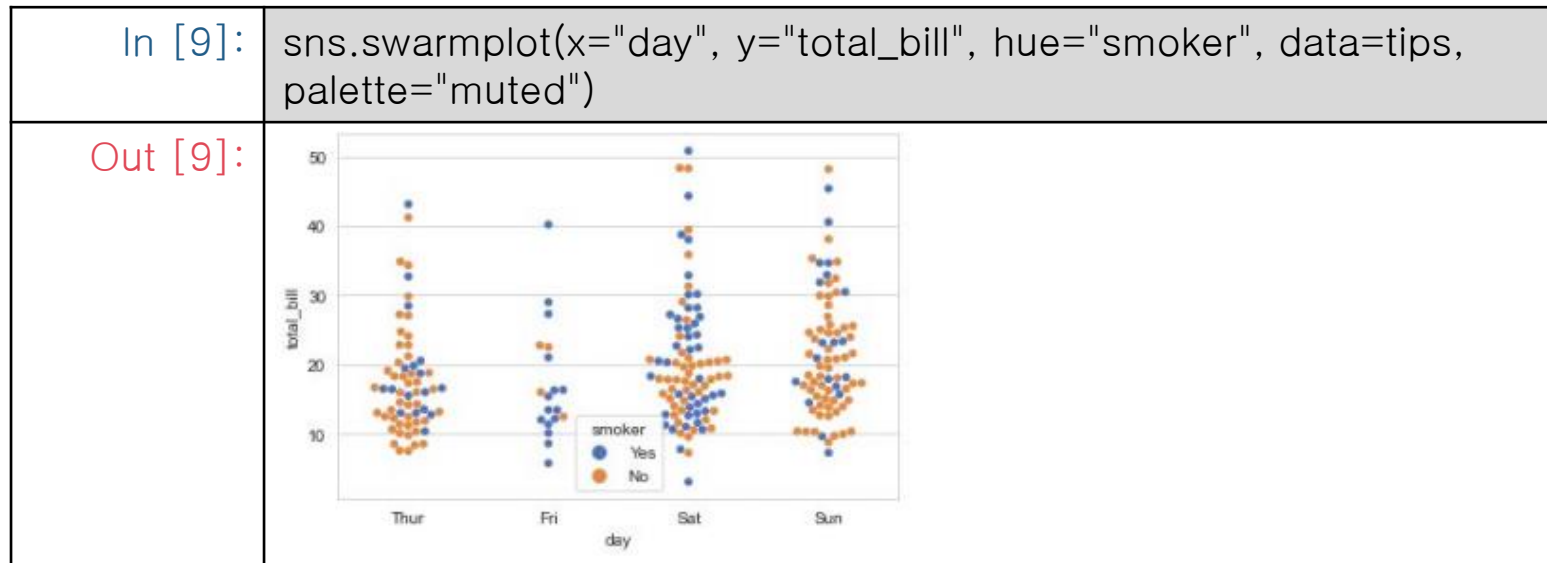
```
sns.violinplot(x="day", y="total_bill", hue="smoker", data=tips, palette="muted")
```

Out [8]:



02 시본

- 스웜 플롯(swarm plot) : 바이올린 플롯과 같은 형태에 산점도로 데이터 분포를 나타냄
- 매개변수 hue로 두 개 이상의 범주형 데이터를 점이 겹치지 않게 정리
 - 영역별 데이터 양을 직관적으로 보여줌



02 시본

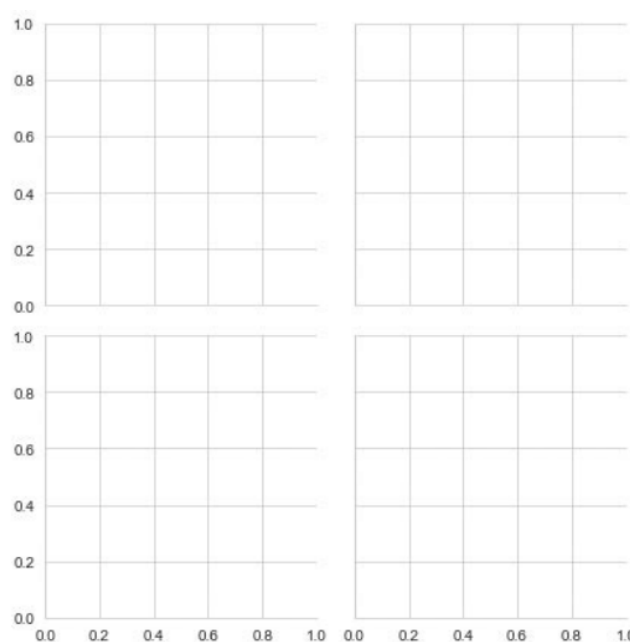
3.2 다양한 범주형 데이터를 나타내는 패싯그리드

- 패싯그리드(FacetGrid) : 그래프의 틀만 제공하여 적당한 그래프를 그려주는 클래스

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

그림 5-5 다음 코드에서 다룰 데이터

02 시본

In [10]:	<pre>g = sns.FacetGrid(tips, col="time", row="sex")</pre>
Out [10]:	

- 기본적인 데이터 표현 틀을 만듦
- 매개변수 col과 row에 범주형 데이터를 넣으면 데이터 종류만큼 'm×n'의 그래프 틀 생성

02 시본

- 그리드가 생성된 후 맵(map)을 사용하여 그래프 만듦
- 각 FacetGrid에 있는 개별 그래프 영역에
그래프를 집어넣는 구조
- 전체 데이터를 범주형 데이터의 다양한 관점에서 나눠서 볼 수 있음

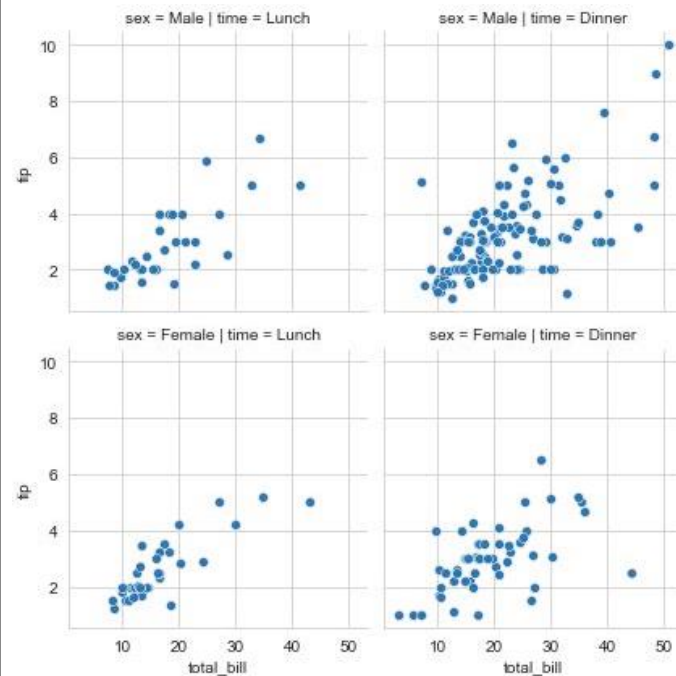
02 시본

In [11]:

```
g = sns.FacetGrid(tips,  
                  col="time",  
                  row="sex")
```

```
g.map(sns.scatterplot,  
      "total_bill", "tip")
```

Out [11]:

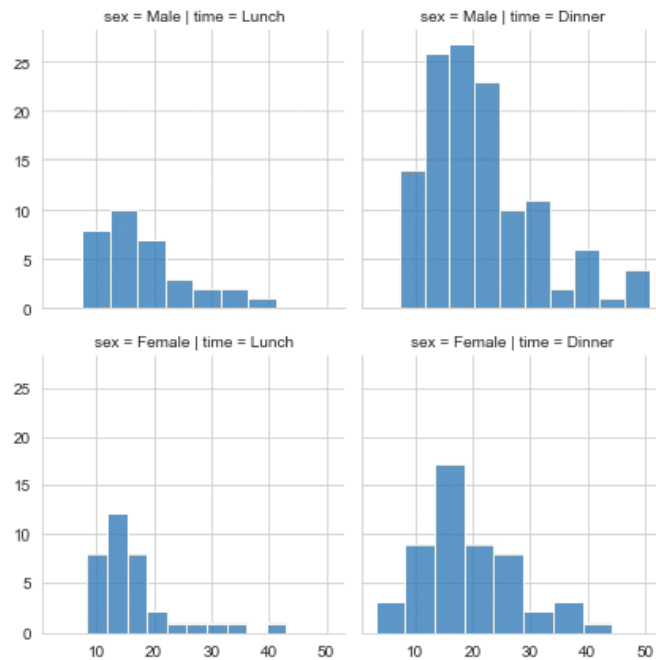


02 시본

In [12]:

```
g = sns.FacetGrid(tips, col="time",  
                  row="sex")  
  
g.map_dataframe(sns.histplot,  
                x="total_bill")
```

Out [12]:



플롯리



03 플롯리

1. 플롯리의 특징

- 플롯리(plotly) : 비즈니스 인텔리전스(Business Intelligence) 대시보드로 개발된 도구
 - 비즈니스 인텔리전스 : BI 도구라고도 불림
사내 여러 데이터들을 정리하여 의사결정을 도움
- 애플리케이션으로, 사용자에게 그래프를 제공
 - 맷플롯립이나 시본은 데이터 분석가들이 데이터의 형태나 분포를 살피기 위해 코드로 사용하는 도구
- 인터랙션 그래프를 지원
 - 인터랙션 그래프 : 그래프 생성 이후 사용자가 인터페이스를 통해 조절 가능

03 플롯리

2. 플롯리 사용하기

- 플롯리 설치
 - 터미널에 명령어 입력

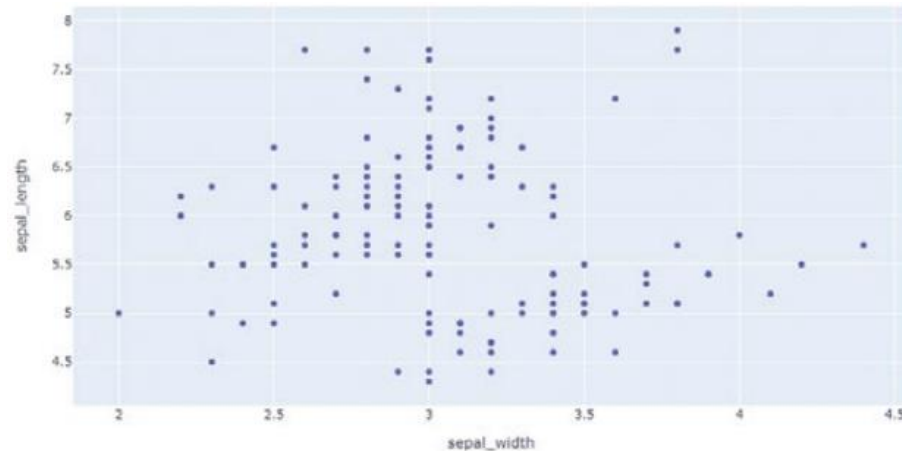
```
(ml) C:\W...>conda install -c plotly plotly
```

- 문법은 맷플롯립이나 시본과 유사

03 플롯리

```
In [1]: import plotly.express as px  
df = px.data.iris() # iris는 판다스 데이터프레임  
fig = px.scatter(df, x="sepal_width",  
                 y="sepal_length")  
fig.show()
```

Out [1]:



- iris 데이터셋을 호출하여 간단한 그래프를 생성
- 래퍼 모듈인 express를 호출한 뒤 산점도를 호출

03 플롯리

- 생성된 그래프에 마우스 커서를 올리면 데이터를 볼 수 있음 (인터랙션 그래프)

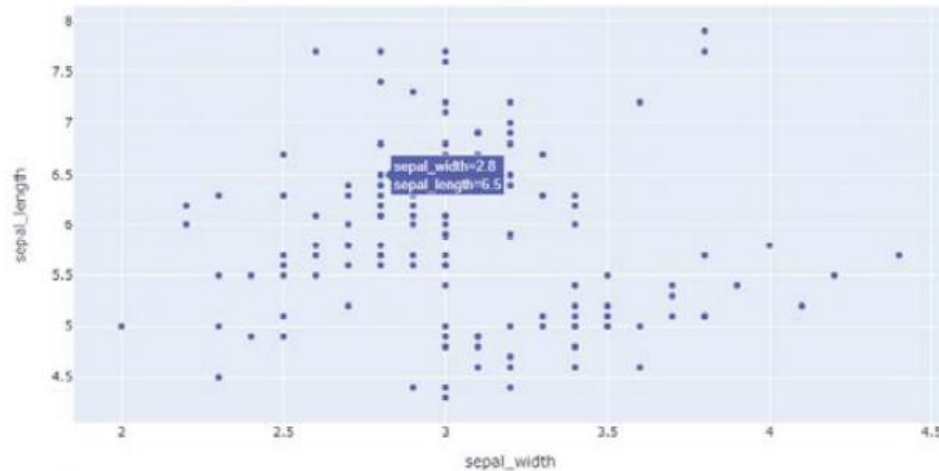


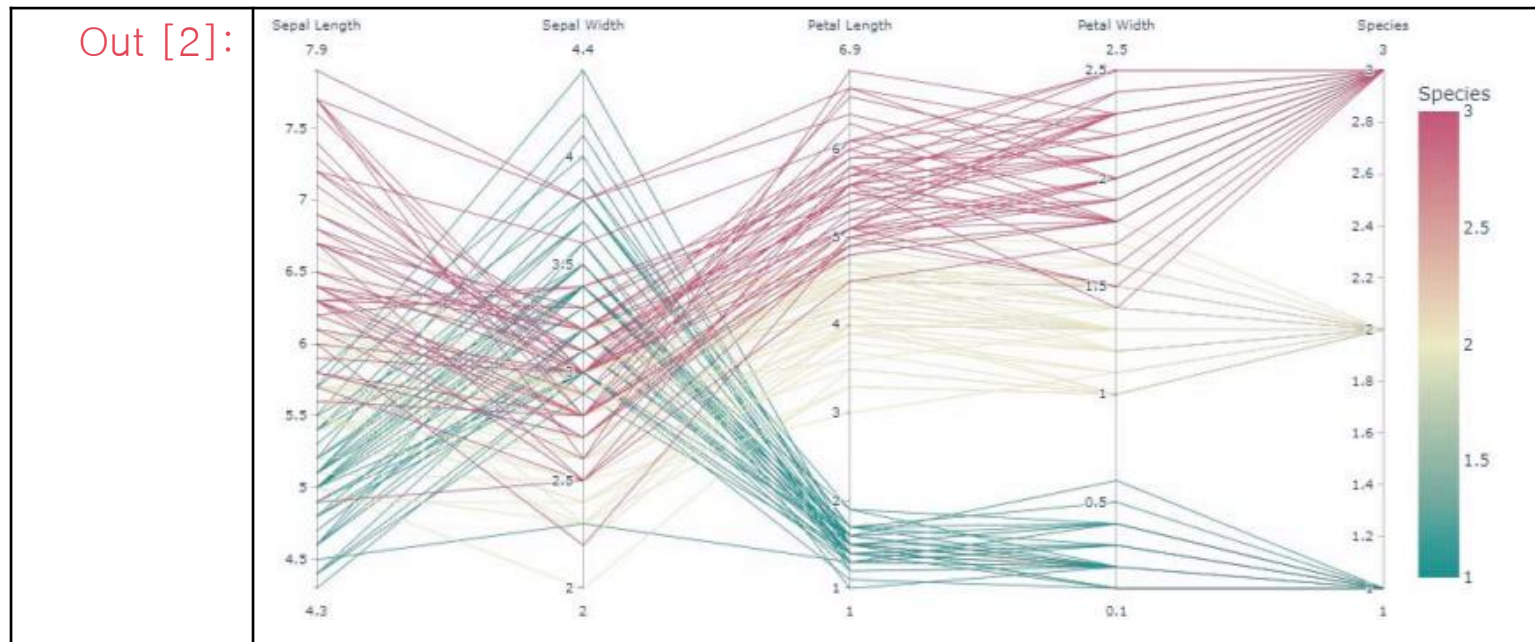
그림 5-6 플롯리로 호출한 그래프의 인터랙션 반응

03 플롯리

- 좌표 그래프(coordinates plot) : 데이터 간 관계를 표현
 - 시본은 제공하지 않지만 플롯리에서 제공하는 기능

```
In [2]: fig = px.parallel_coordinates(df, color="species_id",  
                                     labels={"species_id": "Species",  
                                              "sepal_width": "Sepal Width",  
                                              "sepal_length": "Sepal Length",  
                                              "petal_width": "Petal Width",  
                                              "petal_length": "Petal Length", },  
                                     color_continuous_scale=  
                                         px.colors.diverging.Tealrose,  
                                     color_continuous_midpoint=2)  
fig.show()
```

03 플롯리



[TIP] 다양한 그래프들을 맷플롯립처럼 바닥부터 작성할 수도 있다.
대표적으로 `graph_objects`를 사용하면 다양한 그래프를 만들어 낼 수 있다.